processing speed and therefore often used for such applications requiring only small amounts of data as the delivery of a private key and the execution of digital signature. A typical public key cryptosystem is RSA (Rivest-Sharmir-Adleman). RSA uses a product of very large two prime numbers (for example, 150 digits) to make it difficult to perform factorization (and discrete logarithm) on the product.

Beginning at page 4, line 11:

While RSA cryptosystem based on factorization into prime factors (and discrete logarithm) has sub-exponential decryption, elliptic curve logarithm is considered to have only exponential decryption. While the key size of RSA cryptosystem based on discrete logarithmic problem is 512, 1024, or 2048 bits, the key size of ECC is 160, 192, or 224 bits, which provides generally the same level of security as that of RSA with a shorter key size, resulting in enhanced processing speed.

Beginning at page 5, line 9:

The following describes a public key certificate with reference to FIG. 1. A public key certificate is issued by a certificate authority (CA) or an issuer authority (IA) in the public key cryptosystem. The public key certificate is prepared by a user submitting his ID and public key to a certificate authority and this certificate authority then attaching its ID, validity and signature to the information submitted by the user.

Beginning at page 10, line 4:

In order for the ECC device 23 and the RSA device 33 shown in FIG. 2 to verify the validity of the public key certificate of each other, a configuration must be used where the ECC device 23 and the RSA device 33 send the public key certificates received from each other to the ECC registration authority 22 and the RSA registration authority 32 and then to the ECC certificate authority (ECC-CA) 21 and the RSA certificate authority (RSA-CA) 31 respectively. Inquiries are executed between the ECC certificate authority (ECC-CA) 21 and the RSA

A4 cont'd certificate authority (RSA-CA) 31, and the inquiry results are sent to the devices in place of certification.

Beginning at page 16, line 12:

A5 Preferably, in the above-mentioned public key certificate issuing method, at least one of the plurality of certificate authorities executes processing of storing a generated signature and signature information including signature algorithm information associated with the generated signature into an extended area of the public key certificate.

Beginning at page 25, line 1:

A6 In order to perform cross-certification based on public key cryptosystem, the RSA device 41 and ECC device 42 send their public key certificates to each other. Each device selects, from the plural signatures in the public key certificate received from the other device, the signature which is based on the algorithm which can be verified by it's own device, verifies the selected signature to confirm the validity of the received public key certificate, takes out the public key of the other device from the public key certificate, and executes a cross-certification procedure.

Beginning at page 26, line 2:

A7 The following describes the details of a public key certificate which is applicable to the present invention. A public key certificate is verified by a third party, namely the certificate authority, the issuer of the public key certificate, that a public key for use in the transfer of encrypted data based on the public key or the cross-certification between two parties which transfer data is the public key of the authorized users. The detailed configuration of the public key certificate used in the system of the present invention will be described with reference to FIGS. 5 and 6. An exemplary format of a public key certificate is based on public key certificate format X.509 V3.

Beginning at page 27, line 9:

A8 "validity" is a field in which the starting date and time and ending date and time of the

A8 Concl.

public key certificate are recorded.

Beginning at page 30, line 4:

A9

"Certificate Revocation List Distribution Points" is a field in which a reference point to a revocation list is used for checking if the certificate to be used by user is revoked or not.

Beginning at page 30, line 10:

A10

It should be noted that, in the configuration according to the present invention, not only one signature based on one signature algorithm but also two or more signatures based on different signature algorithms are attached to each public key certificate. This multi-signature configuration will be described later herein.

Beginning at page 31, line 12:

A11

The following describes a method of obtaining a hash value by use of a hash function. A hash function compresses an inputted message to data having a predetermined bit length and outputs the compressed data as a hash value. A hash function is characterized by the fact that it is difficult to infer an input from a hash value (the output) and, when one bit of the data inputted in the hash function changes, many bits of the hash value change, and it is difficult to find different input data that have the same hash value. Hash functions may include MD4, MD5, and SHA-1. DES-CBC may also be used. In this case, MAC (check value, which is equivalent to ICV) providing the final output value becomes a hash value.

Beginning at page 33, line 23:

A12

In step S16, point P = (Xp, Yp) = h1 × G + h2□Ks × G by use of computed h1 and h2. Since the digital signature verifier knows public keys G and Ks × G, scalar multiple of the point on elliptic curve can be computed as with step S4 shown in FIG. 7. In step S17, it is determined whether or not point P is an infinite point. If the decision is No, the procedure goes to step S18 (actually, the decision of infinite point can be done in step S16; namely, is addition of P = (X, Y) and Q = (X, - Y) is made, λ cannot be computed, so that P + Q results in an infinite point). In

A12 cont. step S18, Xp mod r is computed to be compared with digital signature data c. Finally, if a match is found in this comparison, the procedure goes to step S19 to determine that this digital signature is valid.

Beginning at page 35, line 1:

A13 If, in step S20, the digital signature is found invalid, it indicates that the data has been tampered or the entity holding the private key corresponding to the public key has not generated this digital signature.

Beginning at page 36, line 4:

A14 Referring to FIG. 10B, in step S33, hash function h is applied to message M to be verified to generate $m = h(M)$. In step S34, it is determined whether or not $m = S^e \bmod n$ is established. If the decision is Yes, it is determined that the signature is valid in step S35.

Beginning at page 36, line 9:

A15 If the signature is found valid, it indicates that the data has not been tampered, which in turn indicates that the entity holding the private key corresponding to the public key has generated the digital signature.

Beginning at page 36, line 13:

A16 If, in step S34, $m = S^e \bmod n$ is found not established, then, in step S36 it is determined that the signature is invalid, which indicates that the data has been tampered or this digital signature was not generated by an entity which holds the private key corresponding to the public key.

Beginning at page 45, line 9:

A17 Next, the certificate authority (ECC-CA) generates a digital signature from the data (or a message) of the public key certificate on the basis of ECC algorithm. The generated digital signature is stored in the subject Directory Attributes field of the extended area. When the

A17 Cncl. signature generation and storage processing have been completed, the certificate authority (ECC-CA) sends the public key certificate to the certificate authority (RSA-CA) in step S305.

Beginning at page 48, line 24:

As described, the public key certificate stores plural digital signatures based on different signature algorithms, so that the signature verification of the public key certificate can be performed by any of the signature algorithms used. In the above-mentioned processing example, A18 only two signature algorithms are described. It will be apparent that digital signatures based on more than two signature algorithms can be generated and stored. Each certificate authority (XXX-CA) may execute signature generation on the basis of a signature algorithm (XXX) of that certificate authority, or transfer the signed public key certificate to other certificate authorities for sequential signing of this public key certificate.

Beginning at page 49, line 14:

Thus, generating a public key certificate having digital signatures based on different signature algorithms allows two devices each being capable of processing only one ECC and A19 RSA algorithm to execute public key certificate signature verification of each other in cross-certification.

Beginning at page 50, line 5:

First, an exemplary signature verification processing is described with reference to the flowcharts shown in FIGS. 16 and 17 in which the device can process only one signature A20 algorithm, RSA or ECC for example, by use of a multi-signed public key certificate having the format shown in FIG. 12 in which at least the second signature is stored in the extended area and the third, fourth, and so on may be stored.

Beginning at page 51, line 9:

A21 In step S501, the device receives a multi-signed public key certificate (as shown in FIG. 12) from the mate of the communication with which to execute cross-certification. Receiving the

A21 cont. public key certificate, the device identifies the signature algorithm used on the basis of the data stored in the signature.algorithm Identifier field of the basic area to verify whether the processing (or the verification) can be executed by this device in step S502.

Beginning at page 51, line 18:

If the processing (or the verification) can be executed by this device, then, in step S503, this device executes signature verification by applying the signature algorithm recorded to the A22 signature.algorithm Identifier field of the basis area. On the other hand, if this device cannot execute the processing (or the verification), then, in step S504, the device determines by the flag in the extended area of the public key certificate whether there is a digital signature (a second signature) based on another signature algorithm.

Beginning at page 52, line 8:

On the other hand, if the flag indicates the storage of a second signature (for example, the flag = 1), the device references the data in the subject Directory Attributes field in the extended A23 area which stores the signature information about the second signature to determine whether this signature algorithm can be processed by this device in step S505. If the signature algorithm can be processed by this device, then, in step S506, this device executes signature verification by applying the verification algorithm based on the signature algorithm recorded to the subject Directory Attributes in the extended area.

Beginning at page 52, line 20:

If the signature algorithm indicated by the signature information about the second signature is found unexecutable by this device in step S505, then the procedure goes to step S508, in which the device determines whether there is a digital signature based on another signature A24 algorithm. This verification references the third and fourth signature information. As with the second signature information, the device references the subject Directory Attributes field of the extended area.

Beginning at page 54, line 10:

A25

In step S601, a device receives a multi-signed public key certificate (FIG. 13) from the mate of communication with which cross-certification is to be made. Receiving the public key certificate, the device checks the signature algorithm on the basis of the data stored in the signature.algorithm Identifier field in the basic area of the public key certificate to determine whether the processing (the verification) can be executed by this device in step S602.

Beginning at page 55, line 6:

A26

In step S701 shown in FIG. 19, a device receives a multi-signed public key certificate (FIG. 13) from the mate of communication with which cross-certification is to be made. Receiving the public key certificate, the device checks the signature algorithm on the basis of the data stored in the signature.algorithm Identifier field in the basic area of the public key certificate to determine whether the processing (the verification) can be executed by this device in step S702.

Beginning at page 56 line 6:

A27

On the other hand, if the flag indicates the storage of a second signature (for example, the flag = 1), the device references the data in the subject Directory Attributes field in the extended area which stores the signature information about the second signature to determine whether this signature algorithm can be processed by this device in step S705. If the signature algorithm can be processed by this device, then, in step S706, this device executes signature verification on the signature stored in a signature field (a second signature field) other than the basic area and the extended area by applying the verification algorithm based on the signature algorithm recorded to the subject Directory Attributes in the extended area.

Beginning at page 56 line 20:

A28

If the signature algorithm indicated by the signature information about the second signature in step S705 is found unexecutable by this device, then, in step S708, the device determines whether there is still another signature based on another signature algorithm. This

A24 cont. verification references the third and fourth signature information. As with the second signature information, the device references the subject Directory Attributes field of the extended area.

Beginning at page 61 line 3:

A29 One approach to solving the problems of security assurance and enhanced computing speed is the use of a hardware security module (HSM) in holding the signature keys (or private keys) and executing signature processing. Because the HSM is highly tamper-resistant, the use of the HSM plays a significant role in security level enhancement. On the other hand, however, the encryption algorithms to be executed on the HSM are fixed, making it difficult to execute signature key holding and signature processing with other signature algorithms.

Beginning at page 62 line 4:

A30 Each of the registration authorities (RAs) 751 through 755 specifies RSA cryptosystem or ECC cryptography as its permitted signature algorithm for the public key certificates to be issued to the end entities (EEs) under its management. Each registration authority sends to the certification authority (CA) 700 a request for issuing a public key certificate signed on the basis of one or more specified signature algorithms. Each of the registration authorities (RAs) 751 through 755 requests the issuing of a public key certificate signed on the basis of an encryption algorithm which can be processed at the end entities (EEs) managed by this registration authority, namely the encryption algorithm which can be verified at the end entities. Therefore, each of the registration authorities (RAs) desires to have a different signature algorithm.

Beginning at page 64 line 10:

A31 Examples of signature processing to be executed by the signature modules 702a through 702n include RSA, ECC, and DSS (Digital Signature Standard). Further, the computing speed and security level of the signature processing depends on the key length applied in each cryptosystem. The key lengths include 512 bits, 1024 bits, and 2048 bits for RSA, 160 bits, 192 bits, and 224 bits for ECC, each being currently in use. With ECC, in elliptic curve $y^2 = x^3 + ax + b$ on field F(p) (where, p is prime number or exponent of 2), an algorithm for signature

A31 Concl.

processing is determined by characteristic p of field, orders r, a, and b, and base point (Gx, Gy) on the curve, on which the security level also depends.

Beginning at page 65 line 13:

Referring to FIG. 22, there is shown a block diagram of a device configuration of an end entity (EE). As shown, the device has a communication block 831 for executing communication with other devices, content providers, and registration authorities (RAs), an upper controller 832 for controlling the data input/output processing of the entire device, input means 833 including a mouse and keyboard, display means 834 such as CRT or LCD, an encryption processing block 810 for executing signature verification, certification, encryption, and decryption, an external memory 835 for storing the key information for use in content encryption and decryption, and a mass storage block 836 for storing the public key certificate of this device, the public key certificates of service providers, and encrypted content.

A32

Beginning at page 67 line 11:

The mass storage block 836 also stores the public key certificates of service providers, content providers, and other devices with which this device communicates. These public key certificates are also issued by certificate authorities and have each at least one certificate authority signature. In addition, the mass storage block 836 stores encrypted content and registration information.

A33

Beginning at page 67 line 21:

The storage block 812 in the encryption processing block stores a device identifier (ID), a device-unique private key, other private keys, for example, private keys for use as an encryption key in common key cryptosystem, a certificate authority public key for use in verification of public key certificate, a service provider public key for decrypting encrypted data provided by service providers, and a checksum for use as verification data associated with the data stored in the external memory.

A34

Beginning at page 69 line 7:

As described and according to the public key certificate issuing system, public key certificate issuing method, information processing apparatus, information recording medium, and program storage medium associated with the present invention, a novel configuration is provided in which, a public key certificate storing plural signatures based on different signature algorithms such as RSA and ECC are issued and each device selects a signature which can be processed (namely, verified) by itself and verifies the selected signature. Consequently, the novel configuration allows the devices each being capable of verifying only a different signature algorithm to verify the public key certificates of the other devices, so that each device can perform public key certificate verification in the cross-certification and encrypted data communication not only with the other devices having public key certificates attached with signatures based on the same signature algorithm as that of each device, but also with the other devices or providers having public key certificates attached with signatures based on different signature algorithms from that of each device, thereby significantly enhancing the reliability in communication.

A35